Milad Mikeal

10/20/18

CS 162

Project 3

Design:

- Main
    - Declare variables
        - Selection – main menu option
        - Damage – damage dealt by attacker
        - Again = 1 – To optionally rerun program
        - Round = 0 – Keep track of rounds
        - Choice1 & choice2 – Choose characters
    - Create character pointers
    - Create main menu and add options
        - 1. Start
        - 2. Quit
    - Create characters menu & add options
        - 1. Vampire
        - 2. Barbarian
        - 3. Blue Men
        - 4. Medusa
        - 5. Harry Potter
    - Create play again menu
        - 1. Yes
        - 2. No
    - Prompt user w/ main menu
        - If user opts to start game
        - Loop while ‘again’ = 1
            - Prompt character menu for first character
            - Prompt character menu for second character
            - Set values of character
            - List characters and strength points
            - Loop while characters not dead
                - Increment round
                - Attack with player 1
                - Defend with player 2
                - If player 2 still alive
                - Attack with player 2
                - Defend with player 1
                - Declare winner when a character dies
                - Ask user if they would like to play again

- Character
  - Declare necessary getters/setters
  - Declare virtual functions
  - Set virtual constructor to default
- Vampire
  - Inherit from character
  - Armor = 1, strength = 18
  - Attack
    - Virtual
    - Based on one 12-sided die
  - Defense
    - Virtual
    - 50% chance to charm (special ability)
    - Based on one 6-sided die
- Barbarian
  - Inherit from character
  - Armor = 0, strength = 12
  - No special
  - Attack
    - Virtual
    - Based on two 6-sided dice
  - Defense
    - Virtual
    - Based on two 6-sided dice
- Blue Men
  - Inherit from character
  - Armor = 3, strength = 12
    - Number of defense die based on remaining strength
    - 1 - 4 strength = 1 die
    - 5 - 8 strength = 2 dice
    - 9+ strength = 3 dice
  - Attack
    - Virtual
    - Based on two 10-sided die
  - Defense
    - Virtual
    - Determined by remaining strength
- Medusa
  - Inherit from character
  - Armor = 3, strength = 8
  - Attack
    - Virtual
    - Based on two 6-sided dice

            - Automatic kill if roll is 12 (special ability)
        - Defense
            - Virtual
            - Based on one 6-sided die
- Harry Potter
    - Inherit from character
    - Armor = 0, strength = 10/20 (special ability)
        - Start with 10 strength points
        - Revive once with 20 strength points when first set runs out
    - Attack
        - Virtual
        - Based on two 6-sided dice
    - Defense
        - Virtual
        - Based on two 6-sided dice

Main Menu Test Table: 1 – Start, 2 – Quit

Test Case	**Input**	**Expected Outcome**	**Outcome**
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 2	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input = 1 or Input = 2	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

Characters Choice Menu:

Test Case	**Input**	**Expected Outcome**	**Outcome**
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 6	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input between 1 - 6	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

Play Again Menu: 1 – Yes, 2 – No

Test Case	**Input**	**Expected Outcome**	**Outcome**
Input too low	Input < 1	Prompt user to re-enter value	User prompted to re-enter value
Input too high	Input > 2	Prompt user to re-enter value	User prompted to re-enter value
Input in range	Input = 1 or Input = 2	Proceed accordingly depending on user's input	Program proceeded accordingly
Non-numeric input	Input = alpha	Prompt user to re-enter value	User prompted to re-enter value

Reflection:

This project seemed a little challenging at first, but it turned out to be not so bad. I really enjoyed it, and I look forward to building on this project later on.

I did run into a few hiccups along the way. The first was a minor logic bug. For some reason, in each of my defense methods, I had the local defense roll subtracting the input parameter, which happened to be other character's attack roll. It should have been the other character's attack roll subtracting the defense roll. Obviously, that was an easy fix, I just switched them around.

Another problem I ran into was that my character pointers were initially calling the pure virtual functions instead of overriding them with each character's appropriate method. After a bit of tinkering here and there, I solved the problem by dynamically allocating the character types. In doing so, I ran into another small problem. My program worked fine, but I was getting error messages when it came to my delete statements. After a little research, I solved this by declaring a virtual destructor and setting it to default.

Despite all the problems, the project was a very fun one. Through all my initial mistakes, I learned a lot. After all my tests, it turns out Blue Men are more powerful than any other character. Both their attack roll and initial defense roll are way too high for any other character. The next most powerful character is Harry Potter. His Hogwarts ability just makes him outlive any character besides Blue Men. Vampire usually beats Barbarian due to his charm ability and high strength. Medusa is the weakest character. Her glare special is just not that effective; it only has a 1/36 chance.